Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1965-01-01

A computer approach to resource allocation within the framework of C.P.M. scheduling.

Horacek, Jerry L.

Massachusetts Institute of Technology

http://hdl.handle.net/10945/11830

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS ARCHIVE
1965
HORACEK, J.

A COMPUTER APPROACH TO RESOURCE ALLOCATION WITHIN THE FRAMEWORK OF C. P. M. SCHEDULING

JERRY L. HORACEK

LIBRARY
U.S. NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

95

A COMPUTER APPROACH TO RESOURCE ALLOCATION WITHIN THE FRAMEWORK OF C.P.M. SCHEDULING

by

JERRY L. HORACEK

B.S., UNITED STATES NAVAL ACADEMY
(1959)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
in
CIVIL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JANUARY, 1965

Signature of Author ______________________________
DEPARTMENT OF CIVIL ENGINEERING
JANUARY, 1965

Certified by ______________________________
THESIS SUPERVISOR

Accepted by ______________________________
CHAIRMAN, DEPARTMENTAL COMMITTEE
on GRADUATE STUDENTS

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NPS ARCHIVE
1965
HORACEK, J.

~~Thesis
H757~~

Library
U.S. Naval Postgraduate School
Monterey, California

ACKNOWLEDGMENTS

The author wishes to express his most sincere appreciation to Professor Albert Dietz for his very helpful advice, suggestions and assistance.

In addition the author is further indebted to LT. Robert A. Schade, CEC, USN and LTjg Salvatore A. Martinelli, CEC, USN whose helpful comments and reviews proved most useful.

The author especially thanks his wife, Judy, for her inspirational aid and needed encouragement as well as her assistance in proof-reading and competent typing of the manuscript into its final form.

The facilities and services of the M.I.T. Computation Center were utilized for work reported herein.

ABSTRACT

A COMPUTER APPROACH TO RESOURCE ALLOCATION WITHIN THE FRAMEWORK OF C.P.M. SCHEDULING

by

JERRY L. HORACEK

Submitted to the Department of Civil Engineering on 18 January, 1965
in partial fulfillment of the requirements for the degree of Master
of Science.

The Critical Path Method of scheduling has become widely accepted by the construction industry as the most efficient method available for the scheduling of construction projects. Although this method attempts to arrive at the schedule of least cost, based upon individual activity estimates of time duration and cost; it does not take into consideration the relationships that exist between the various activities and their respective requirements for the same resource simultaneously. This of course leads to excessive fluctuations in the daily resource requirements of the project, which in turn may make the C.P.M. schedule economically not feasable.

In this thesis a new attempt is made at solving the resource problems by allocating resources to the individual activities on the basis of their relative need for a resource. The restrictions established by the C.P.M. schedule, as well as activity resource requirements are taken into consideration in such a manner that the float or slack time available to each activity is utilized in the most judicious manner to achieve the minimum of daily resource fluctuations for the project.

Because of the complexity of the algorithm demonstrated in this paper, a computer program was developed to utilize this allocation technique. The program is demonstrated on sample problems, and its effectiveness is thoroughly analized.

Thesis Supervisor: Albert G. H. Dietz

Titles: Professor of Civil Engineering and
Professor of Architecture

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION. 1

CHAPTER 2 - THE CRITICAL PATH METHOD OF SCHEDULING. 4

CHAPTER 3 - RESOURCE SCHEDULING 11

CHAPTER 4 - A RESOURCE ALLOCATION TECHNIQUE 15

CHAPTER 5 - ALLOCATION RESTRICTIONS AND LOGIC 22

CHAPTER 6 - THE COMPUTER APPROACH 30

CHAPTER 7 - CONCLUSIONS 42

BIBLIOGRAPHY. 61

LIST OF FIGURES

FIGURE 2-1 5
FIGURE 2-2 10
FIGURE 4-1 21
FIGURE 5-1 23
FIGURE 5-2 24
FIGURE 5-3 26
FIGURE 6-1 35
FIGURE 6-2 37
FIGURE 6-3 37
FIGURE 7-1 45

APPENDICES

A - PROGRAM LISTING. 46
B - COMPUTER PROGRAM VARIABLES 57

BIBLIOGRAPHY 61

A COMPUTER APPROACH TO
RESOURCE ALLOCATION
WITHIN THE FRAMEWORK OF C.P.M.
SCHEDULING

CHAPTER 1

INTRODUCTION

Prior to World War II, the planning and scheduling of construction projects was based primarily on past experience and guesses, but studies conducted during the war years tended to show that a more quantitative approach to the problems of planning and scheduling in all industrial fields could be achieved. For the construction industry, the most significant break-through came with the development of a network flow theory by the E. I. du Pont de Nemours Company. This theory came to be known as the Critical Path Method of scheduling.

Almost simultaneously with the development of C.P.M., other scientific break-throughs occured in the field of electronic data computers. Refinements and developments on both C.P.M. theory and computer design by many parties have led to C.P.M. computer programs which today have become widely accepted and used by the construction industry.

Today a construction firm theoretically can obtain economically and quickly the most profitable schedule for any project. Never the

less, several major problems have yet to be solved before construction scheduling can truely be termed a scientific procedure. The scheduling of resources to a project and the resultant time duration estimates of specific items of work are conducted in a manner which has changed very little over the years. This has lead to probably the greatest criticism of the use of C.P.M. More specifically, it is argued that any quantitative procedure which relies so heavily on such qualitative input data is of limited practical value.

A reliable resource allocation technique appears to offer the key to a completely quantified approach to the scheduling and planning problem. To be considered successful, such a technique should have a closer relationship with other scheduling aids (e.g.,C.P.M.) than presently exists. In addition, the results of the allocation should be acceptable from the economic view point. This objective can be achieved only when the day to day fluctuations of resource usage are reduced to a minimum.

It is the objective of this thesis to develop a computerized resource allocation technique which will fulfill the following requirements:

1. The technique must have a close relationship with present C.P.M. techniques by not only following the basic restrictions set up by the C.P.M. schedule, but also by providing data for the updating of the C.P.M. schedule. This feedback will in most cases take the form

of more accurate activity time duration estimates than the original ones.

2. The technique must allocate resources in a manner that is based on a nearly constant resource usage each day of the project. Economically this is a much more feasible method than the previous attempts of maintaining resource usage below a preset maximum amount.

With such a technique, it is envisioned that construction management personnel will not only be able to plan and schedule projects more quantitatively but will also be able to update these schedules effectively with heretofore unused feedback from the field.

CHAPTER 2

THE CRITICAL PATH METHOD OF SCHEDULING

In this section, the subject of C.P.M. will be discussed. However, it is neither the desire nor the intent of the author to define or explain in its entirety the much discussed subject of C.P.M. Rather, C.P.M. will be presented in this paper only to the extent necessary to aquaint the reader with the problems facing any attempts at the allocation of resources. With this in mind, the only divisions of C.P.M. that will be examined (to any extent) are the project modeling and the project analysis phases.

For a more complete knowledge of C.P.M. the reader is referred to either The Critical Path Method, Its Implementation And Effective Utilization in Construction by Joseph S. Keller (8) or Lecture Notes On Critical Path Scheduling by J. Lloyd Cutcliffe (4).

The project modelling or the "arrow diagram phase" is basically the process of breaking a project into the individual activities and representing them in such a manner that they can be analysised by mathematical methods. The activities are in themselves meaningful items of work, the sum of which, when performed in a certain sequence, make up the project.

The beginning and end of each activity are known as events and considered to be discrete points in time. In the graphical representation of a project, the activities are drawn as arrows, with the arrow-

head signifying completion of an activity and the tail the beginning of that activity. The events are represented by circles known as nodes.

Several activities may share either the same beginning or ending node, but an event cannot take place until all activities leading into it have been completed and no activity may be started until its beginning node has been reached. The overall network of arrows may have only one starting event and one ending event. This is a great aid in computer solution programs

To aid in visualizing the graphical representation of a project, consider the data given below for a simple example problem.

Activity Description	Must preceed	Estimated Duration
a (any meaningful	e,f	2
b item)	c,d	3
c	e,f	2
d	g	2
e	-	5
f	g	3
g	-	4

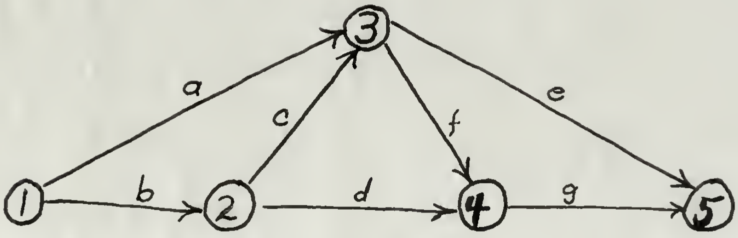


FIGURE 2-1

Although the activities are un-named in this example, it should

be understood that the level of detail implied by the activities is variable and therefore no "one" model represents a given project. The model is in fact extremely flexible, thereby gaining great latitude in representing the structure of the project accurately relative to a particular need or situation. However, since any given operation can be broken up into any number of activities, the reader is warned against the possibility of the diagram becoming confusing with extreme use of detail.

Once the initial project modelling is completed, the static project analysis may begin. In brief, this phase of C.P.M. is the mathematical manipulation of the model to reveal those activities which define total project time. The effect all other activities might have on the project are also revealed at this point. In order to accomplish these items, static analysis utilizes activity duration, or the time it is estimated it will take to complete an activity. It should be noted at this point that in practice the accuracy and reliability of any terms defined in the remainder of this section are based on the accuracy of the estimated activity duration. For the purpose of the techniques used in this paper, all times will be assumed to be in working days.

A number of basic definitions must be made at this point to aid the reader in gaining an insight into the static analysis. For pur-

poses of demonstration, the sample project model shown in Figure 2-1 will be employed.

The earliest event time (EET(n)) (where n is the node number) is defined as the earliest point in time that event n can occur. Mathematically, if i is the node at the tail of the activity, j is the node at the head of that activity and DUR (i,j) is the duration of that activity:

$$\mathrm{EET}(j) = \text{Maximum } \left[\mathrm{EET}(i)+\mathrm{DUR}(i,j)\right]$$

By the convention used in this paper, the EET of a node is the earliest working day that any activity originating at that node can be started. Thus the EET(1) is equal to day 1.

Example:

$$\mathrm{EET}(3) = \text{Max of} \left\{ \begin{array}{c} \left[\mathrm{EET}(1)+\mathrm{DUR}(1,3)\right] \\ (1+2) \\ 3 \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{c} \left[\mathrm{EET}(2)+\mathrm{DUR}(2,3)\right] \\ (4+2) \\ 6 \end{array} \right\}$$

$$= 6 \text{ days}$$

The EET is calculated for all nodes beginning with the start of the project and ending with the terminal node.

Conversly, the latest event time (LET(n)) is defined as the latest point in time that event n can occur. LET's are calculated in a manner similar to EET except that the terminal node is computed first and calculations are continued until the origin is reached.

Mathematically:

$$LET(i) = \text{Minimum} \quad \left[LET(j) - DUR(i,j)\right]$$

Example: Assuming that LET (5) and LET (4) have previously been calculated to be 13 and 9 days respectively, then;

$$LET(3) = \text{Min of} \left\{ \begin{array}{c} \left[LET\ (5) - DUR\ (3,5)\right] \\ \\ (13-5) \\ \\ 8 \end{array} \right\} \text{ or } \left\{ \begin{array}{c} \left[LET\ (4) - DUR\ (3,4)\right] \\ \\ (9-3) \\ \\ 6 \end{array} \right\}$$

$$= 6 \text{ days.}$$

In the scheduling of activities, four terms which can be derived from combinations of activity duration, and event times are of special interest. Early Start (ES) and Early Finish (EF) are the earliest points in time an activity can be respectively started and finished if all preceeding activities are completed as scheduled. Conversly, Late Start (LS) and Late Finish (LF) are respectively the latest points in time an activity can be started or finished without delaying the scheduled completion of the project. Mathematically these terms are expressed as follows:

$$ES(i,j) = EET(i)$$

$$EF(i,j) = EET(i) + DUR(i,j)$$

$$LS(i,j) = LET(j) - DUR(i,j)$$

$$LF(i,j) = LET(j)$$

In any discussion of C.P.M., one invaritably encounters the term float or slack time. This is basically the extra time allowed in the scheduling of the activity in question, and occures when activities of different time durations are required to terminate at the same node. More specifically, there are three commonly used types of float, each of which is a specific combination of activity duration and the various starts and finishes just defined.

Total float (TF) is equal to the maximum amount of time that the activity can be extended without affecting the scheduled completion time of the project, or mathematically:

$$\begin{aligned} TF(i,j) &= LF(i,j) - EF(i,j) \\ &= LS(i,j) - ES(i,j) \end{aligned}$$

Free float (FF) is equal to the amount of time an activity can be extended without affecting the early start (ES) of any succeeding activity, or mathematically:

$$FF(i,j) = EET(j) - EF(i,j)$$

Independent float is equal to the amount of time an activity can be extended without affecting the early times of succeeding activities or late times of preceeding activities, or mathematically:

$$IF(i,j) = MAX \quad [EET(j)-LET(i) - DUR(i,j)]$$

Any activity which contains no float is called critical. No leeway in the scheduling of a critical activity is allowed and any

increase in the time required will result in a corresponding increase in the duration of the project. When these activities are taken together from origin to terminus, they comprise what is known as the critical path.

Considering the data given in Figure 2-1 once again, a more complete and informative schedule for this project can now be constructed as follows:

(i,j)	Description	DUR(i,j)	ES	LS	EF	LF	TF	FF	IF
1-2	b	3	1	1	4	4	0	0	0
1-3	a	2	1	4	3	6	3	3	3
2-3	c	2	4	4	6	6	0	0	0
2-4	d	2	4	7	6	9	3	3	3
3-4	f	3	6	6	9	9	0	0	0
3-5	e	5	6	8	11	13	2	2	2
4-5	g	4	9	9	13	13	0	0	0

Figure 2-2

CHAPTER 3

RESOURCE SCHEDULING

In the construction of the project model, as discussed in the last section, little consideration was given to the resource requirements for the project. Resources are undoublably in the minds of persons making the activity duration estimates for the schedule, but in all probability this is usually done on an activity by activity basis, with little thought given to the interrelationships that exist among the various activities when taken as a project. Therefore the possibility of excessive resource requirements, on any one day due to the simultaneous need for that resource by several activities, does exist and points to one of the major criticisms of the use of C.P.M. scheduling.

Idealy, most businessmen prefer to keep the resource requirements below a daily maximum level and at the same time maintain the day to day fluctuations at a minimum. The economic implications of not following such a policy are quite apparent. Consider for example the case of requiring a particular trade of workers for a project in accordance with the hypothetical schedule given below:

DAY	NUMBER OF MEN REQUIRED
1	7
2	2
3	15
4	1
5	12
TOTAL	37

Unless the firm has several projects under construction at the same time, the problem of hiring and firing costs, or conversly, the cost of retaining idle personnel is immediately apparent. When dealing with expensive equipment as the resource under consideration, the problem becomes even more acute; and on a long and more complex project, leveling resource usage becomes a very complicated and tedious proposition.

With such an obvious need for a reliable technique of allocation resources in a more desirable manner, it is only natural that a great deal of reasearch has taken place in this field. To date however, only two reasonably successful methods of solution have appeared and neigher of these methods produces what might be considered an optimum solution.

In the techniques which attempt to allocate resource within the structure of a C.P.M. schedule, the most realistic method used is by the manipulation of activity floats. Both of the previously cited methods, commonly called the serial and parallel methods, attempt to utilize this approach. In the serial method, activities are ranked according to ascending node numbers and then, allocated resources one activity at a time. Activities are scheduled at the earliest possible time according to the C.P.M. schedule and the availability of the remaining resources. The resources available are in turn a function of a predetermined resource limit. When the

availability of resources becomes zero, the allocation to that activity is delayed as necessary in order not to exceed the preset limit. There is also a preset limit on the extent to which an activity can be delayed and when this limit is reached, the activity is assigned the required resources regardless of the excessiveness of this maneuver. The major criticism of the serial method is that it does not optimize the proper function in that the daily resource fluctuations, which are highly objectionable, still exist.

The parallel method takes a different approach to the problem by working with a group of activities at one time. Activities are chosen which can proceed simultaneously over a certain period of time without exceeding a set resource limit. This usually begins with the most critical activities and then fits in the remaining activities, depending on the amount of total float that exists for each activity. The obvious drawback to this approach is that, other than by trying an almost infinite number of times, there is no way of determining if the group chosen leads to the optimum solution.

Another important consideration which should be examined at this point is that both methods assume that the daily crew size for each activity is required to remain constant. For example if an activity requires 40 resource days to be completed in 5 days, the crew size each working day is required to be 8 resources. This

would intuitively appear to be a highly objectionable restriction to place on any allocation technique. While it is realized that for some activities this restriction would be necessary, in the majority of cases this is not true and in actuality this restriction is seldom practiced. It would seem a more realistic approach to set a maximum and minimum crew size limit within which the efficiency of work would not noticebly be affected. This approach will be discussed more fully in the next section.

The various other, less effective, techniques which have been developed also use activity floats in a effort to allocate resources in a desirable and realistic manner, but in all cases they are much less successful than the two methods just discussed. In all allocation techniques to date it appears that the predetermined objective functions upon which the optimization of resource usage is based has been poorly chosen. This has resulted in solutions which are not truly beneficial or desirable in the eyes of the prospective user.

which [illegible] appear to be a [illegible] in terms of [illegible]. While it is realized that [illegible] would be necessary, in the majority of cases [illegible] considered. It would seem a more [illegible] approach [illegible] not necessarily be [illegible]. This approach will be discussed more fully in the next section.

The various other, less [illegible], techniques which have been developed also use [illegible] in [illegible] but in all cases they are much less [illegible] than the two methods just discussed. In all [illegible] it appears that the [illegible] have been greatly [illegible]. This has resulted in [illegible].

CHAPTER 4

A RESOURCE ALLOCATION TECHNIQUE

Prior to an explanation of the technique used in this paper while striving to achieve a better solution of the resource allocation problem, the basic objectives and assumptions will be discussed. As pointed out in the preceding section, the major dissatisfaction that seems to exist with all of the available allocation methods is that the end result is a schedule which is very unrealistic and of little practical value to the construction industry. It is hypothesized that the error lies not so much with techniques used as it does in the basic assumptions made. A prime example of this is the basic assumption that a crew size must remain constant. Although this may be a desirable aim, it should not be an over-riding restriction. In actual practice, a crew size is often increased or decreased if the activity is proceeding behind or ahead of schedule.

Another restriction placed on past techniques is that they were forced to be very elementary in form to avoid becoming too arduous for hand solutions. With the increased use of electronic digital computers, this restriction can be discarded; and iterative techniques or other time-consuming methods can now be employed when necessary.

Once these major obstacles to the resource allocation problem have been recognized and proven to be of little actual significance,

a more sensible approach can be taken toward achieving a realistic solution. In particular, more vigorous methods can now be employed and thus a more ambitious objective pursued. A stabilized resource usage, or at least one with extremely dampened fluctuations is the proposed objective of the allocation technique presented in this paper.

While trying to formulate such an objective within the framework of a C.P.M. schedule, several basic assumptions were made and will be discussed at this point. A variable sized activity crew, since it exists in the actual world, was not deemed objectionable and therefore was reasoned to be a plausible solution.

A second, probably more disputable, assumption is that each item of work can be started and stopped within the limits set up by the C.P.M. schedule and the schedule of activity maximum-minimum resources. While it is recognized that in some cases such an assumption would prove unacceptable, it is also a proven fact that most construction work can be delayed a day or so after once being initiated (such as is often done over weekends, holidays or in cases of bad weather) with no ill effects. The entire assumption can thus be described as a value judgement made by the author as to the relative merit of having an assumption which would, in a small minority of cases, result in an unacceptable solution; but one which, in the larger majority of cases, would provide not only a solution which was totally

acceptable but would also bring a dynamic flexibility to the method used to gain that solution. Once such an assumption was made the algorithm was no longer restricted to merely altering activity times within narrow limits but would allow more plausible methods of allocating on the basis of activity resource exigencies.

The most judicious method of allocation investigated was one based upon a ratio of resource availability to resource requirement for each activity. This mode effectively establishes an activity's actual resource needs. Utilizing the C.P.M. schedule, the availability of resources for an activity was determined to be the total amount of time available for the completion of an activity times the maximum number of resources per day that would compose an efficient force. It is obvious that this factor will diminish as the number of days available gets smaller or as the number of resources available for the day in question becomes smaller.

The denominator of this ratio is merely the required resource days necessary to complete the activity, as estimated in the original C.P.M. data. Quite understandably, this factor remains constant until a resource is allocated to the activity, at which time it too is decreased.

The following example serves to illustrate how this ratio would be determined for those activities given in Figure 2-2 which are capable of receiving resources on the 1st working day.

All resource data used in the example can be assumed part of the estimate used to formulate the C.P.M. schedule.

DAY 1

Activity	Late Finish	Max Resources/Day	Total Resources Required
1-2	4	3	9
1-3	6	3	6

$$\text{Ratio} \quad \frac{(\text{LF-DAY})(\text{MAX Res/Day}) - (\text{Resources Allocated this Date})}{(\text{Total Resource Required at this Time})}$$

Activity	Ratio
1-2	$\frac{(4-1)(3)-0}{9} = 1$
1-3	$\frac{(6-1)(3)-0}{6} = \frac{15}{6}$

Assume the allocation of 1 resource day to activity 1-2:

Activity	New Ratio
1-2	$\frac{(4-1)(3)-1}{9-1} = 1$
1-3	$\frac{(6-1)(3)-1}{6} = \frac{14}{6}$

An interesting point to observe in the above example is that critical activities, by their very nature of requiring the maximum efficient crew size, will always have a ratio of one as long as they are allocated resources first. Non critical activities will have a ratio which is greater than one, and any activity that is "super critical" will have a ratio less than one.

The above observations bring forward two important concepts to the issue of resource allocation. First of all for such a ratio to be accurate, it must be computed after each allocation of one resource per day. With the advent of electronic computers this of course poses no great problem for the desired technique. The more important aspect is that, with such a ratio, critical activities can be immediately recognized and resources allocated to these activities. At the same time, activities whose requirements can not possibly be fulfilled (i,e. supercritical activities) can also be recognized and appropriate steps can be taken to alleviate this situation.

Although the system of allocation of the individual activities has been resolved, the far more complicated problem of determining the proper number of resources to be used each day of the project, remains to be analyzed. After much experimentation, it was determined that the best solution could be found by trial and error methods, using an iterative procedure. This procedure will be described in the following paragraphs and will refer to Figure 4-1 for descriptive purposes.

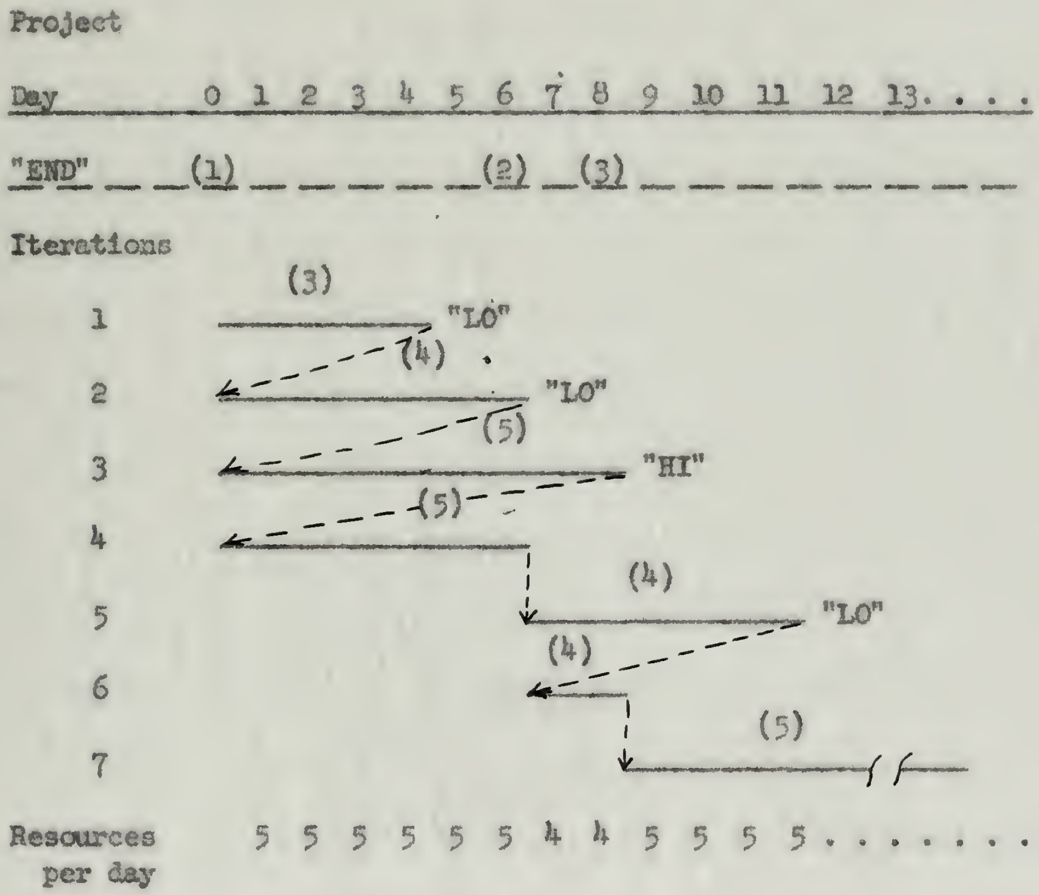
Prior to beginning the procedure, two initialization restrictions must be fulfilled. First, the primary "END" is designated as day 0. The term "END" is used to inform the algorithm how far to backtrack before beginning another iteration, as the algorithm is designed to

start each iteration at day (END 1). Therefore as long as "END" is equal to day 0, all iterations will begin at day 1. An example of this is shown in each of the first 4 iterations of Figure 4-1.

The second requirement is that an initial number of project resources per day must be designated. Although this may be any arbitrary amount, much time will be saved if the number chosen is close to the number finally used. For this reason, it is recommended that the average of the total number of resource days required by the project be used. In Figure 4-1 it is equal to 3.

Much like other network flow algorithms, the allocation method tries to find the number of resources per day that will satisfy all restrictions of the various activity starts and finishes (e.g. early start, early finish, etc.), activity maximum and minimum resources restrictions, and still will provide the longest continuous flow through the network. To accomplish this, the algorithm, once an iteration has stopped, must be able to recognize whether it should add or subtract from the available number of resources per day; and then decide how many days the operation should be backtracked in order to over come the obstacle that halted the procedure.

FIGURE 4-1


Project
Day 0 1 2 3 4 5 6 7 8 9 10 11 12 13. . . .
"END" (1) (2) (3)
Iterations
1 (3) "LO"
2 (4) "LO"
3 (5) "HI"
4 (5)
5 (4) "LO"
6 (4)
7 (5)
Resources per day 5 5 5 5 5 5 4 4 5 5 5 5 . . . .

FOOTNOTE

(4)	4 Resources to be allocated each day.
"LO"	Number of resources/day not sufficient to meet requirement on day indicated by "LO".
"HI"	Number of resources/day too many to satisfy requirement for day indicated by "HI".

CHAPTER 5

ALLOCATION RESTRICTIONS AND LOGIC

There are many restrictions the algorithm must fulfill before an allocation can be made or an iteration continued. For example, an activity can not be considered for allocation purposes unless the day of the iteration lies between the C.P.M. solution for Early Start and Late Finish for that activity. Also the total resource requirements of any one activity may not be fulfilled prior to the Early Finish date of that activity. When the latter case does arrise, it serves to prove that too many project resources per day are being allocated to the activities under consideration. The converse of this is of course true if the total resource requirements of an activity are fulfilled after the Late Finish date of that activity.

Figure 5-1 is presented to show the relationship between the various symptoms and causes of restriction violation. In Figure 4-1 a surplus of project resources per day is designated as "HI" and a scarcity is designed as "LO".

As one might expect, there are cases where if both the input parameters and restrictions remain constant, the algorithm would tend to go into a computational loop. For example if an iteration is stopped at day 2 because of a scarcity (LO) of project resources/day and after starting a new iteration at the beginning, is stopped again

SYMPTOM	CONSTRAINT VIOLATED	CAUSE
Activity considered for allocation prior to C.P.M. Early Start. Activity total resource requirement fulfilled prior to C.P.M. Early Finish.	C.P.M. Schedule	Preceding activity ended too soon, i.e., too many project resources/-day.
Activity considered for allocation after C.P.M. Late Start. Activity total resource requirement not fulfilled by C.P.M. Late Finish.	C.P.M. Schedule	Preceding activity ended too late, i.e., not enough project resources/-day
Project Resources remain unallocated after all activities under consideration have been allocated the max. daily resource limit.	Activity Maximum Resource Limit	Too many project resources/day available.
Activity becomes super critical after procedure has fulfilled activity min. resource requirements.	Activity Minimum Resource Limit	Too few project resources/day available.

FIGURE 5-1

at day 4 because of a surplus (HI), something must be done to stop the procedure from becoming a never ending repetition. Obviously, restrictions can not be violated if the technique is going to assume the required orderliness, and therefore the remaining parameter to alter is the date at which an iteration begins. There are two cases when such an operation becomes a necessity; the case described above and the case where a surplus (HI) is reached earlier than a scarcity (LO) is reached. Both cases are treated in a similar manner as depicted in figure 5-2.

Cause of Loop	Solution
LO < HI	1. Assign project resources/day that achieved HI date and begin iteration. 2. When day equal to LO date has finished allocation to activities under consideration: a. Make new "END" date equal to LO date b. Reduce project resources/day by 1 c. Continue Iteration
HI < LO	1. Assign project resources/day that achieved LO date and begin iteration 2. When day equal to HI date has finished allocation to activities under consideration: a. Make new "END" date equal to LO date b. Increase project resource/day by 1. c. Continue Iteration.

Figure 5-2

A graphical illustration for a case of LO < HI is shown in iterations 4-5 of Figure 4-1 and the case of HI < LO in iterations 6-7.

Because the technique logic may be quite difficult to visualize at this point, the reader is advised that a pictorial presentation of

the complete procedure is given in the form of a flow diagram in Figure 5-3 with the accompanying explanation given as the next topic of discussion. In an effort to make the operation of the algorithm less confusing to the reader, an attempt is made to take a broader view of the overall procedure with less emphasis on the micro-operations.

To begin the technique, a search is made of the various activities, as given by the C.P.M. schedule, to determine those activities that can be accomplished simultaneously beginning at the first working day of the project. By the convention used in this paper, this includes only those activities whose beginning node is labeled (1). The next step is to compute a resource necessity rate, in the manner previously outlined, for those activities selected, and then to pick the activity which has the lowest rate. Once this is accomplished, several checks must be performed to insure the activity can receive additional resources for that day without violating any overriding restrictions. If the lowest rate has a value of zero, which means the activity will not use any of the type resource being allocated, a test must be made to insure that activities following it on the C.P.M. diagram are not considered for future allocations prior to their early starts. Also, if the activity chosen has already received its maximum allocation for that day, further allocations to it would

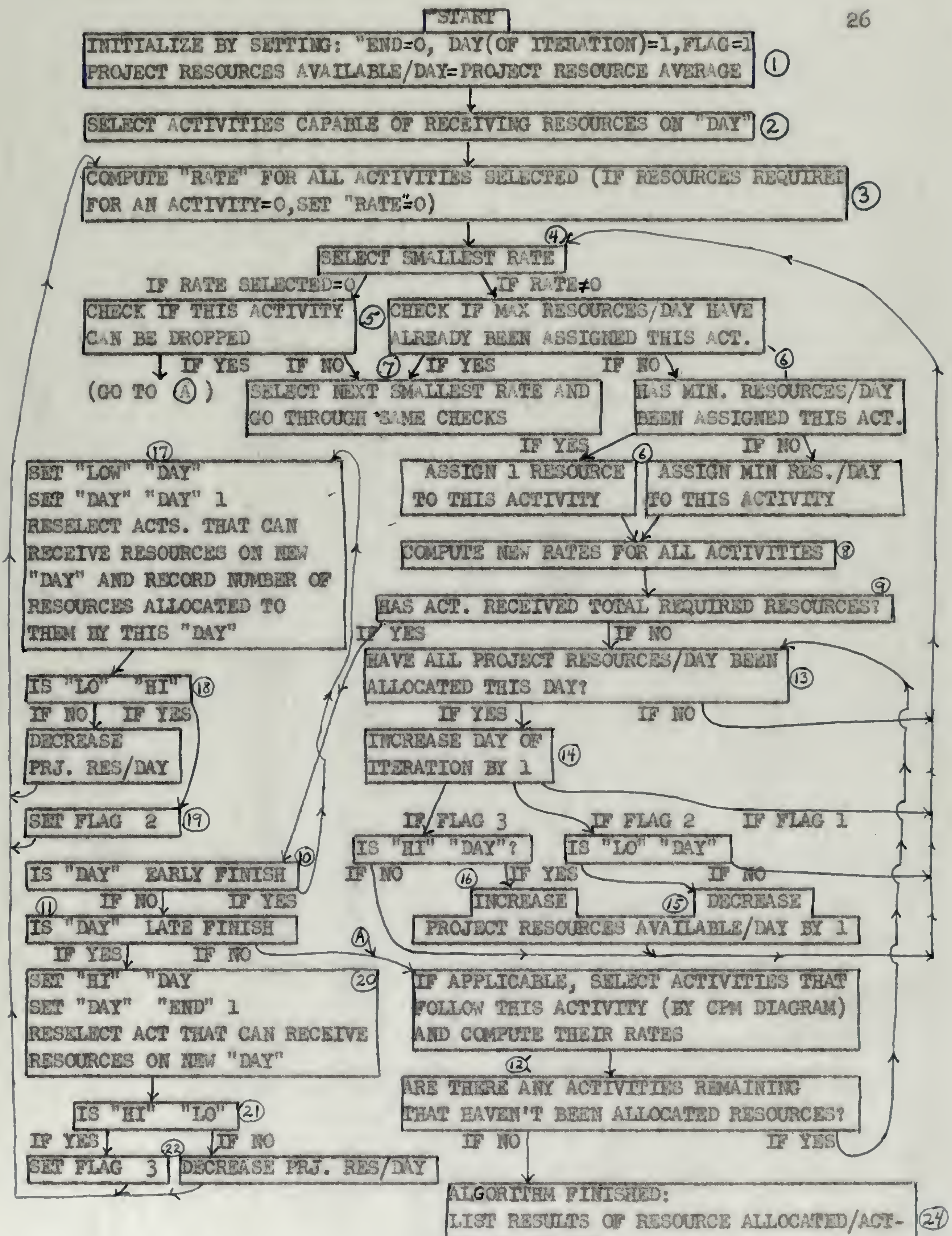

START
INITIALIZE BY SETTING: "END=0, DAY(OF ITERATION)=1,FLAG=1
PROJECT RESOURCES AVAILABLE/DAY=PROJECT RESOURCE AVERAGE
①
SELECT ACTIVITIES CAPABLE OF RECEIVING RESOURCES ON "DAY"
②
COMPUTE "RATE" FOR ALL ACTIVITIES SELECTED (IF RESOURCES REQUIRED FOR AN ACTIVITY=0,SET "RATE"=0)
③
④
SELECT SMALLEST RATE
IF RATE SELECTED=0
IF RATE≠0
CHECK IF THIS ACTIVITY CAN BE DROPPED
⑤
CHECK IF MAX RESOURCES/DAY HAVE ALREADY BEEN ASSIGNED THIS ACT.
IF YES
IF NO
⑦
IF YES
IF NO
⑥
(GO TO Ⓐ)
SELECT NEXT SMALLEST RATE AND GO THROUGH SAME CHECKS
HAS MIN. RESOURCES/DAY BEEN ASSIGNED THIS ACT.
IF YES
IF NO
⑥
ASSIGN 1 RESOURCE TO THIS ACTIVITY
ASSIGN MIN RES./DAY TO THIS ACTIVITY
COMPUTE NEW RATES FOR ALL ACTIVITIES
⑧
⑨
HAS ACT. RECEIVED TOTAL REQUIRED RESOURCES?
IF YES
IF NO
HAVE ALL PROJECT RESOURCES/DAY BEEN ALLOCATED THIS DAY?
⑬
IF YES
IF NO
INCREASE DAY OF ITERATION BY 1
⑭
IF FLAG 3
IF FLAG 2
IF FLAG 1
IS "HI" "DAY"?
IS "LO" "DAY"
IF NO
IF YES
IF NO
⑯
INCREASE
⑮
DECREASE
PROJECT RESOURCES AVAILABLE/DAY BY 1
⑰
SET "LOW" "DAY"
SET "DAY" "DAY" 1
RESELECT ACTS. THAT CAN RECEIVE RESOURCES ON NEW "DAY" AND RECORD NUMBER OF RESOURCES ALLOCATED TO THEM BY THIS "DAY"
IS "LO" "HI"
⑱
IF NO
IF YES
DECREASE PRJ. RES/DAY
SET FLAG 2
⑲
IS "DAY" EARLY FINISH
⑩
⑪
IF NO
IF YES
IS "DAY" LATE FINISH
IF YES
IF NO
Ⓐ
SET "HI" "DAY
SET "DAY" "END" 1
RESELECT ACT THAT CAN RECEIVE RESOURCES ON NEW "DAY"
⑳
IF APPLICABLE, SELECT ACTIVITIES THAT FOLLOW THIS ACTIVITY (BY CPM DIAGRAM) AND COMPUTE THEIR RATES
⑫
ARE THERE ANY ACTIVITIES REMAINING THAT HAVEN'T BEEN ALLOCATED RESOURCES?
IF NO
IF YES
IS "HI" "LO"
㉑
IF YES
IF NO
㉒
SET FLAG 3
DECREASE PRJ. RES/DAY
ALGORITHM FINISHED:
LIST RESULTS OF RESOURCE ALLOCATED/ACTIVITY, BY DAY
㉔

FIGURE 5-3

lead to the violation of a different restriction. In either case, if an activity is found unsuitable for consideration, another check of the list must be made to select the next lowest rate. This procedure is followed until a suitable activity is discovered; and then the allocation procedure continues.

Once the proper activity is determined, the next check is to determine the number of resources that are to be assigned to that activity. If the allocation is the first of the day, the activity will be assigned the minimum daily efficient size force, otherwise it will be allocated just one resource.

The next step is to recompute a new rate for each activity under consideration, noting whether any activities have received their total requirement for the job. If an activity has, checks are then made to insure that too many or too few project resources per day were not utilized and if so the procedure for this problem and the redesignation of the successful "END" day are accomplished in the manner previously explained. If these checks are successful, another test must be made to ascertain if any activities still under consideration end with the same node number. This is done because of the C.P.M. restriction that no activities commencing at a node can be started until all activities terminating at that node have been accomplished. If it is plausible to consider following activities, the technique

makes the previous end node the beginning node, and all activities commencing at this point are selected and added to the list of activities under consideration for resources.

At this point, the paths again converge and the procedure becomes the same, whether an activity has received its total project requirement or not. Logically the next test is made to determine if all the project resources for the day under consideration have been previously allocated. If they have, the day of iteration is increased by one, and checks are made for a new designation of the "END" point as well as required increases in the allowed number of project resources per day as previously explained. This series of tests terminates one iteration so the procedure goes back to the point of selecting the smallest rate and starts again.

Since the only time the overall technique may be completed is after the last activity has received its alloted number of resources, a check is made each time an activity accomplishes this feat to determine if all other activities have been previously considered. If no activities remain to be considered, the iterative procedure is terminated and a listing is made of the assigned resources by activity number, date of allocation, and daily amounts of allocation to that activity.

It must be remembered that the just described procedure considers

only one of the many different types of resources that have to be allocated on a project. However, when considering the resources to be allocated over the length of any project, one type invariably stands out as being more critical, as far as usage or availability, than any of the other types under consideration. With this in mind, a feasible technique to employ in establishing a firm time schedule for allocating the various types of resources is to allocate the critical resource first and let that schedule establish the absolute limits for the activity times to be used in allocating the remaining types of resources. Thus after the allocation of the first, or critical, resource is completed, none of the original C.P.M. floats would exist between activities; but all future allocations would have the flexibility of being accomplished within the boundaries established by the critical resource. In this manner it is insured that only one C.P.M. schedule need be used by all the resources allocated to a project.

If more than one resource appears critical, several separate solutions for the project must be achieved, each using one of the aforementioned resources as the critical one for that solution. The final complete solutions are then compared to find the most acceptable overall solution.

CHAPTER 6

THE COMPUTER APPROACH

As one can realize from reading the preceding chapter, the amount of data that would have to be calculated and retained in the process of accomplishing the technique described in this paper could be overwhelming. Consider for example a project of 25 or 30 activities extended over a period of 50 working days and requiring about 500 resource-days total. This would in fact be comparable to a small sized construction project. If, using the technique described in the last chapter, by some quirk of fortune, the algorithm was successful through the whole project with its first choice of resources per day to allocate, it would be a conservative estimate to say a minimum of 10 thousand separate calculations and comparisons would have to be conducted. This, coupled with the multitude of intermediate data required for calculation and thus from necessity retained, as well as the final answers which must be recognized and retained, gives some insight into the book-keeping problems that would exist if the soulution were calculated by hand. This is of course for a highly idealized situation which in all probability would never happen, but this fact tends to show that a hand solution would be even more tedious and mistake-prone. As pointed out previously, the answer to this problem is quite obviously the utilization of electronic computers.

Once this problem has been resolved, the next question to be answered is what will the computer program be comprised of and what input data will be required to achieve a satisfactory output. As a partial answer to this question, some of the various programs written to solve the critical path problem were investigated and it was found that such programs were both quite numerous and effective. For this reason it was felt that any attempt to offer a new C.P.M. program would result in something of insiginficant value. J. S. Keller (8) also pointed out a fact of much importance at this point, by showing that a much more flexible approach to the scheduling problem could be achieved by utilizing a modular approach employing program packages. Ideally then, the solution lies in making the allocation program compatible with one of the existent C.P.M. programs.

For this reason, Keller's Schedule and Float Computing Routine (JKESLS) was chosen as the C.P.M. program that would partially provide input data for the allocation program (JHREAL). The output of the C.P.M. program provides activity numbers, activity durations, early and late starts, early and late finishes, and the various floats. Thus, the only additional data that must be provided as input for JHREAL are the activity maximum and minimum daily resource limits and the total resource requirements for each activity as well as a starting value for project resources available each day. These

additional values, in all probability, form an important part of the original estimate of activity duration and therefore would be no great problem to obtain on an actual construction project. In fact once the estimator is required to furnish these data in addition to the activity duration, the management may find that the estimates are more accurate than they had been previously.

Another reason the Keller C.P.M. program was chosen for input is the fact that JKESLS was written to be used on a timesharing routine. This of course implies that a large computer would be utilized. Because of the memory required either in core storage or on tapes to employ JHREAL, a computer with extensive memory capabilities is a mandatory requirement if the project size is to be of any consequence. For this reason and in a effort to make the program compatible with JKESLS, the allocation technique was programed in FORTRAN for the IBM 7094 computer.

It was discovered, however, that the facilities for Time-Sharing were not available for use by the author. Therefore, in order to actually test JHREAL on a computer it was written without the time sharing capabilities. The alterations necessary to make JHREAL completely compatible with Keller's C.P.M. Program packages and Time Sharing are not considered great however, and therefore it is urged that this possibility be investigated further by anyone interested in the actual implimentation of JHREAL.

One further point should be discussed before delving into the

actual computer program, and this is the fact that it appears that an unlimited number of daily resources is available for allocation. In a sense this is true since the program will not stop or print an error statement if the technique allocates more than the maximum daily number the user has in mind. This was done purposely in order that the user could see what he actually needs in the way of resources for his present C.P.M. schedule. This allows the user the opportunity to either pick a less than the optimum C.P.M. schedule or accept the allocation as it is, using overtime or other necessary measures in order to meet the allocation schedule. It also gives the management a chance to decide if the estimators are being too conservative on their estimates. If this is the case, the whole project should be resubmitted after the necessary changes are made.

Once the basic ground rules for the computer program have been explained, the intricacies of the program itself can be pursued. The reader will find two aids extremely helpful in the program description. The first of these is the MACRO flow diagram (figure 5-3). As a further aid to the reader, each block of the flow diagram contains a number which corresponds to the number shown on the listing, immediately prior to the section of the program that applies to its respective block. Appendix A contains a complete annotated listing of the computer program. The reader will also find Appendix B, which contains defini-

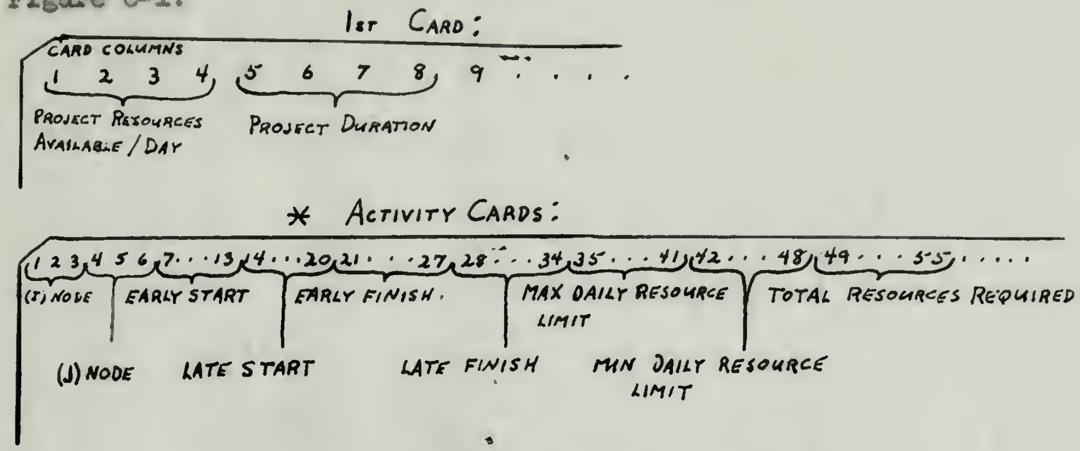
tions of the program variables, extremely useful if the program listing is going to be read and understood.

The last chapter explained the need for initial set up conditions for the technique and the same needs are true for the computer program. These will cause no problem to the user however, since they form an integral part of the program, as can be seen by comparing the block numbers previously mentioned.

Due to lack of time sharing facilities and, therefore, JKESLS output, all input from JKESLS to the allocation program is simulated using punched cards. There are two basic forms used in the formation of these cards; one giving a starting value for the daily project resources available and for project duration; the second form listing each activity separately with its starts, finishes, floats and daily activity resource limits. After all activity cards are punched, an additional card of the same form is made, but a negative activity number is used. This card signals the computer that there is no more data to be read.

Rather than going into a long diseration about the various modes of printing input data, it will suffice to say that all values must be right adjusted in their allotted series of columns. Thus if four card columns are allotted for a value and only 3 will be used for that value, the left most column will be left blank. If this rule is followed no decimal points are used. The allocation of columns is pictured in

Figure 6-1.



*NOTE: After all activity data has been recorded, a modified last card is required and it must contain a negative (I) node number.

FIGURE 6-1

Sample input data and the corresponding output results appear in Figures 6-2 and 6-3. Because the time available for testing the program was some what limited, only relatively small, hypothetical projects were used for tests. It is felt however the two projects shown in this paper present allocation problems which are typical of those found in actual practice. It must be realized however that no computer program can be considered completely debugged and tested until it has been employed for a considerable period of time.

An interesting sidelight of the examples shown is that the input data for the example shown in Figure 6-3 was taken from Richards S. Moore's Thesis (11), and the output are the results from the computer program outlined in this paper. Moore's technique is a hand computed procedure which attempts to level resources usage.

Although it will be discussed more fully in the next chapter, it is worth mentioning that such a comparison was considered of considerable interest and value by the author.

FIGURE 6-2

SAMPLE INPUT DATA AND RESULTANT OUTPUT SCHEDULE OF ALLOCATIONS.

```
*     DATA
   3   5
 1   2   1   1   3   3   4   2   5
 1   4   1   1   4   5   3   2   5
 2   4   3   3   5   5   5   1   4
-1   4   3   3   5   5   5   1   4
```

```
                          SCHEDULE OF ALLOCATIONS

FOR DAY  1  ACTIVITY  1- 2  REQUIRES  3 RESOURCES.
                                 TOTAL FOR THIS DAY    3

FOR DAY  2  ACTIVITY  1- 2  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 4  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  3  ACTIVITY  1- 4  REQUIRES  3 RESOURCES.
            ACTIVITY  2- 4  REQUIRES  1 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  4  ACTIVITY  2- 4  REQUIRES  3 RESOURCES.
                                 TOTAL FOR THIS DAY    3
```

FIGURE 6-3

SAMPLE INPUT DATA AND RESULTANT OUTPUT SCHEDULE OF ALLOCATIONS.

```
*     DATA
   6  41
 1   2  1    1    5    6    2    2    8
 1   3  1    1   11   11    0    0    0
 1   4  1    1    3   28    2    1    4
 1   7  1    1    4   35    6    2   18
 2   3  5    6   10   11    8    2   40
 2   6  5    6   13   25    2    2   18
 2   7  5    6   13   35    0    0    0
 3   5 11   11   22   22    0    0    0
 4   7  3   28   10   35    5    2   35
 5   6 22   22   25   25    4    4   12
```

FIGURE 6-3 (CONTINUED)

```
 5   8 22   22   34   41   3   2   36
 6   8 25   25   41   41   2   2   32
 7   8 13   35   19   41   3   2   18
-7   8 13   35   19   41   3   2   18
```

```
                    SCHEDULE OF ALLOCATIONS

FOR DAY  1  ACTIVITY  1- 2  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  3 RESOURCES.
                                 TOTAL FOR THIS DAY    5

FOR DAY  2  ACTIVITY  1- 2  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  3 RESOURCES.
                                 TOTAL FOR THIS DAY    5

FOR DAY  3  ACTIVITY  1- 2  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 4  REQUIRES  1 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  2 RESOURCES.     5
                                 TOTAL FOR THIS DAY

FOR DAY  4  ACTIVITY  1- 2  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 4  REQUIRES  1 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  2 RESOURCES.     5
                                 TOTAL FOR THIS DAY

FOR DAY  5  ACTIVITY  2- 3  REQUIRES  5 RESOURCES.
                                 TOTAL FOR THIS DAY    5

FOR DAY  6  ACTIVITY  2- 3  REQUIRES  7 RESOURCES.
                                 TOTAL FOR THIS DAY    7

FOR DAY  7  ACTIVITY  2- 3  REQUIRES  7 RESOURCES.
                                 TOTAL FOR THIS DAY    7

FOR DAY  8  ACTIVITY  2- 3  REQUIRES  7 RESOURCES
                                 TOTAL FOR THIS DAY    7
```

FIGURE 6-3 (CONTINUED)

```
FOR DAY   9 ACTIVITY  2- 3  REQUIRES  7 RESOURCES.
                                TOTAL FOR THIS DAY    7

FOR DAY  10 ACTIVITY  2- 3  REQUIRES  7 RESOURCES.
                                TOTAL FOR THIS DAY    7

FOR DAY  11 ACTIVITY  1- 4  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  2 RESOURCES.
            ACTIVITY  2- 6  REQUIRES  2 RESOURCES.
                                TOTAL FOR THIS DAY    6

FOR DAY  12 ACTIVITY  1- 4  REQUIRES  2 RESOURCES.
            ACTIVITY  1- 7  REQUIRES  2 RESOURCES.
            ACTIVITY  2- 6  REQUIRES  2 RESOURCES.
                                TOTAL FOR THIS DAY    6

FOR DAY  13 ACTIVITY  2- 6  REQUIRES  2 RESOURCES.
            ACTIVITY  4- 7  REQUIRES  4 RESOURCES.
                                TOTAL FOR THIS DAY    6

FOR DAY  14 ACTIVITY  2- 6  REQUIRES  2 RESOURCES.
            ACTIVITY  4- 7  REQUIRES  4 RESOURCES.
                                TOTAL FOR THIS DAY    6

FOR DAY  15 ACTIVITY  1- 7  REQUIRES  2 RESOURCES.
            ACTIVITY  4- 7  REQUIRES  4 RESOURCES.
                                TOTAL FOR THIS DAY    6

FOR DAY  16 ACTIVITY  1- 7  REQUIRES  2 RESOURCES.
            ACTIVITY  4- 7  REQUIRES  3 RESOURCES.
                                TOTAL FOR THIS DAY    5

FOR DAY  17 ACTIVITY  4- 7  REQUIRES  5 RESOURCES.
                                TOTAL FOR THIS DAY    5

FOR DAY  18 ACTIVITY  4- 7  REQUIRES  5 RESOURCES.
                                TOTAL FOR THIS DAY    5

FOR DAY  19 ACTIVITY  4- 7  REQUIRES  5 RESOURCES.
                                TOTAL FOR THIS DAY    5

FOR DAY  20 ACTIVITY  4- 7  REQUIRES  4 RESOURCES.
            ACTIVITY  7- 8  REQUIRES  2 RESOURCES.
                                TOTAL FOR THIS DAY    6
```

FIGURE 6-3 (CONTINUED)

```
FOR DAY 21  ACTIVITY 4- 7  REQUIRES 4 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 22  ACTIVITY 5- 6  REQUIRES 4 RESOURCES.
            ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 23  ACTIVITY 5- 6  REQUIRES 4 RESOURCES.
            ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 24  ACTIVITY 5- 6  REQUIRES 4 RESOURCES.
            ACTIVITY 5- 7  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 25  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 26  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 27  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 28  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                    TOTAL FOR THIS DAY      6

FOR DAY 29  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6

FOR DAY 30  ACTIVITY 5- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 6- 8  REQUIRES 2 RESOURCES.
            ACTIVITY 7- 8  REQUIRES 2 RESOURCES.
                     TOTAL FOR THIS DAY     6
```

FIGURE 6-3 (CONTINUED)

```
FOR DAY  31  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  7- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    6

FOR DAY  32  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  33  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  34  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  35  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  36  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  37  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  38  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  39  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4

FOR DAY  40  ACTIVITY  5- 8  REQUIRES  2 RESOURCES.
             ACTIVITY  6- 8  REQUIRES  2 RESOURCES.
                                 TOTAL FOR THIS DAY    4
```

CHAPTER 7
CONCLUSIONS

In the preceeding chapters, a new attempt at resource allocation has been explained and demonstrated; However due to the complexity and variability of the allocation problem, many prospective users of such a technique are often disheartened when they discover that it is not a cure-all for every allocation problem that can and does arise. On the contrary, the technique advocated by this paper does have recognized limitations and also extensive capabilities, both of which must be understood if the technique is to be effectively utilized.

First, it is essential that the close interrelationship that must exist between the technique and C.P.M. be realized from the beginning. The results of the allocation program are only as good as as the accuracy of the estimates used to establish the C.P.M. schedule. However, the allocation results can be used as a back check on the original C.P.M. assumptions to insure that they are feasible and accurate. For example, one may find that high resource usage on any one day may be attributed to a faulty duration estimate of one or more activities that are proceeding on that day. Also before accepting a less than optimum C.P.M. schedule, in order that a certain daily resource limit not be exceeded, the cost differential between the

less than optimum C.P.M. schedule and the resource acquisition costs should be thoroughly investigated.

In addition to accurate C.P.M. estimates, another requirement of the allocation technique is that activity divisions be made with care and common sense. This must be accomplished since the allocation technique does not have the capability of recognizing relationships that may exist between the various types of resources utilized. Thus, if such a relationship does exist and is of significant importance on any specific item of work, that item must be divided into more than one activity

The allocation method takes advantage of several assumptions to approach the resource problem in a new manner. The splitting of activities, the concept of variable sized activity crews and the use of an allocation rate based on activity urgency are the major assumptions which lead to a dynamic approach.

Although the program was not used to schedule and actual project, it was compared to a schedule taken from R. S. Moore's thesis (11). A graphical comparison of Moore's results and those obtained from JHREAL are illustrated in Figure 7-1. The comparison of daily project resource usage is highly favorable in that the fluctuations with JHREAL are smaller in amplitude and larger in period that Moore's data. This of course would mean less hiring and firing problems. An added

advantage would be that Moore's method is limited to relatively small projects with only a few limiting restrictions.

Effective as the technique described in this paper may appear, the fact remains that it is still highly limited in its effectiveness by the number of fixed restrictions placed upon it. An example of this is the fact that if one does not desire his activities to be split, the algorithm described in this paper can not be used since it is by fixed restriction that all items may be split within the limits set up by the C.P.M. schedule. It is felt that the secret to success lie in closer man to machine communication. It is hoped therefore that future work in allocation techniques will be in adapting JHREAL for Time Sharing. In accomplishing this, it is emphasized that the real value will lie in a program, a few of whose minor restrictions can be altered as necessary to fit the individual construction project. In fact with the recognized complexity of the problem, the optimum solution may require a new computer language which is tailored exclusively for resource allocation rather than merely writing limited programs in existing languages.

FINAL RESOURCE USAGE

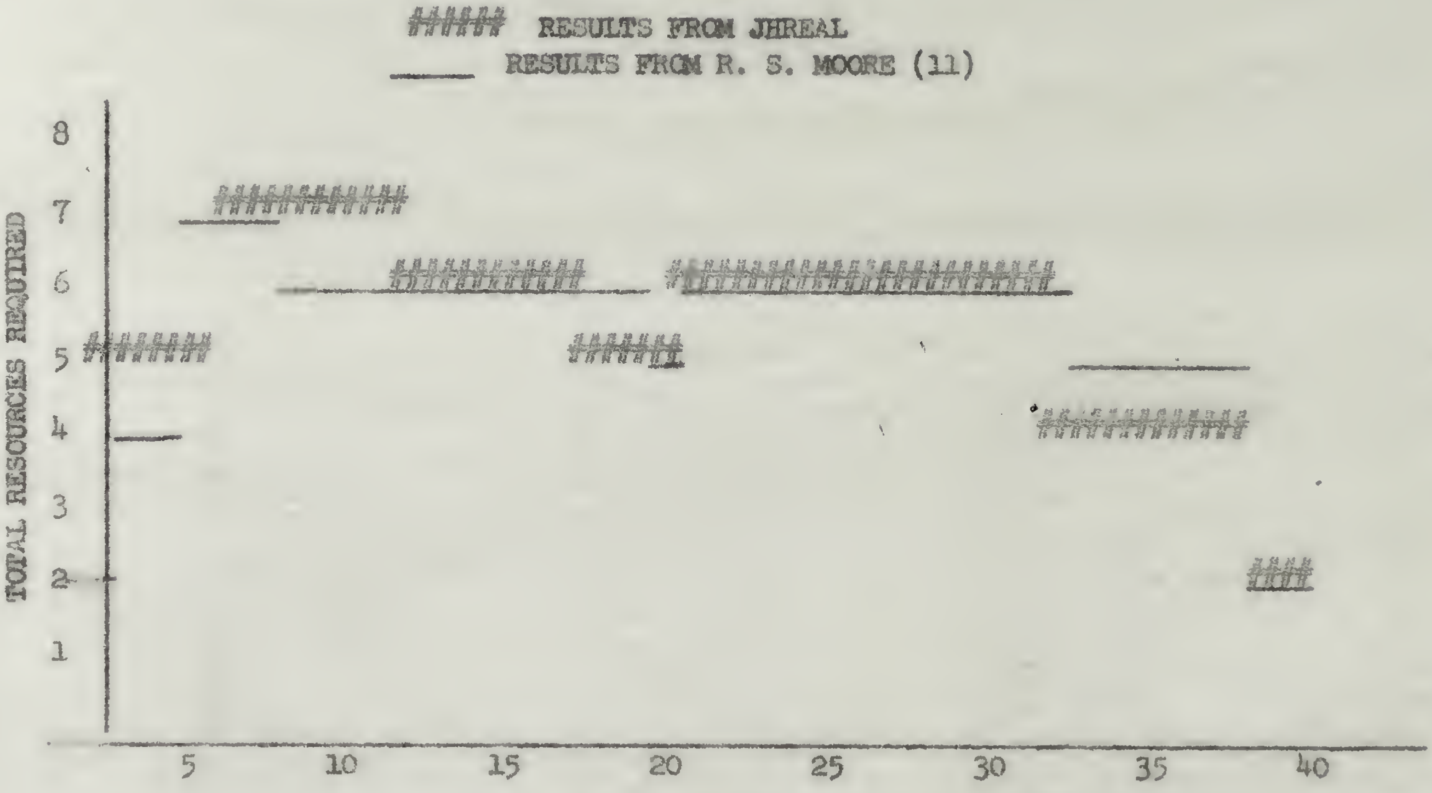


FIGURE 7-1

APPENDIX A

PROGRAM LISTING

```
*     XEQ
*     LIST 8
C     BLOCK 1 FLOW DIAGRAM
C     START OF JHREAL
      REWIND 10
      REWIND 11
C
      DIMENSION I(500),J(500),ES(500),ALS(500),EF(500),ALF(500),RESMAX(5
     100),RESMIN(500),TOTREQ(500),TAVAIL(500),AVAIL(500),DEN(500),RATE(5
     200),ASSIGN(500),JSUCC(500),GOOD(500),ANTER(3),FINIS(2)
C
      COMMON I,J,ES,ALS,EF,ALF,RESMAX,RESMIN,TOTREQ,TAVAIL,AVAIL,DEN,RAT
     1E,ASSIGN,JSUCC,GOOD
C     INITIALIZATION
      ENZLO=.1
      ENZHI=.9
      JA=0
      IBEST=1
      DAY=1.
      N=0
      NEWLOW=1
      JFLAG=1
      ENZ=0.
      NODE=0
      HI=0.
      ALO=0.
C     READ PARAMETER CARD GIVING STARTING DAILY RESOURCE LIMIT AND PRJDUR
      READ 1,RAVAIL,PRJDUR
C     READ ACTIVITY INPUT CARDS AND COUNT NUMBER
    5 N=N+1
      READ 3,I(N),J(N),ES(N),ALS(N),EF(N),ALF(N),RESMAX(N),RESMIN(N),TOT
     1REQ(N)
      ASSIGN(N)=0.
C     BLOCK 2 FLOW DIAGRAM
      IF(I(N)-1)7,9,11
   11 JSUCC(N)=0
      GO TO 5
    9 JA=JA+1
```

APPENDIX A (CONTINUED)

```
      JSUCC(N)=1
      FINIS(1)=N
      FINIS(2)=TOTREQ(N)
      WRITE TAPE 11,FINIS
      GO TO 5
    7 NMAX=N-1
      JMAX=JA
      TORES=RAVAIL
      FINIS(1)=-1.
      WRITE TAPE 11, FINIS
C     BLOCK 3 FLOW DIAGRAM
C     CALCULATE RATE FOR ALL ACTS. WITH I=1
      DO 13 K=1,JMAX
      IF(RESMAX(K) -RAVAIL) 17,19,19
   17 GOOD (K)=RESMAX(K)
      AVAIL(K)=RESMAX(K)
      GO TO 21
   19 GOOD(K)=RAVAIL
      AVAIL(K)=RAVAIL
   21 TAVAIL(K)=(ALF(K)-ES(K))*AVAIL(K)
      DEN(K)=TOTREQ(K)
      IF(TOTREQ(K)) 12,14,12
   14 RATE(K)=0.
      GO TO 13
   12 RATE(K)=TAVAIL(K)/DEN(K)
   13 CONTINUE
C     BLOCK 4 FLOW DIAGRAM
C     TEMP ASSIGNMENT OF SMLF TO 1ST WORKABLE ACT.
  197 DO 23 K=NEWLOW,NMAX
      IF(JSUCC(K)-1) 23,25,23
   25 ICHECK=1
      LOW=K
      SMLF=ALF(K)
      KSML=K
      LOWP1=LOW+1
      GO TO 27
   23 CONTINUE
      SELECTION OF ACTUAL SMLF
   27 IF(LOWP1-NMAX) 28,28,37
   28 DO 29 K=LOWP1,NMAX
      IF(JSUCC(K)-1) 29,31,29
   31 ICHECK=ICHECK+1
      IF(SMLF-ALF(K)) 33,33,35
   35 SMLF=ALF(K)
```

```
      KSML=K
   33 IF(ICHECK-JMAX) 29,37,29
   29 CONTINUE
   37 ICHECK=1
      IBEST=LOW
      SMALL=RATE(IBEST)
C     SELECT SMALLEST RATE AND IT'S SUBSCRIPT
      IF(LOWP1-NMAX) 38,38,47
   38 DO 39 K=LOWP1,NMAX
      IF(JSUCC(K)-1) 39,41,39
   41 ICHECK=ICHECK+1
      IF(SMALL-RATE(K)) 45,45,43
   43 SMALL=RATE(K)
      IBEST=K
   45 IF(ICHECK-JMAX) 39,47,39
   39 CONTINUE
C     BLOCK 5 FLOW DIAGRAM
C     CHECK TO INSURE ACT. WITH SMALL CAN TAKE RESOURCES
   47 IF(TOTREQ(IBEST)) 46,48,46
   48 IF(DAY-EF(IBEST)) 53,50,50
   50 NODE=1
      JSUCC(IBEST)=0
      JAMAX=JMAX-1
      JAP1=IBEST+1
      GO TO 129
C     BLOCK 6 FLOW DIAGRAM
   46 IF(RESMIN(IBEST)-1.) 49,49,51
   49 IF(ASSIGN(IBEST)-RESMAX(IBEST)) 55,53,53
   55 GIVE=1.
      GO TO 57
   51 IF(ASSIGN(IBEST)-RESMIN(IBEST)) 59,49,49
   59 IF(RESMIN(IBEST)-AVAIL(IBEST)) 61,61,53
   61 GIVE=RESMIN(IBEST)
   57 IF(DEN(IBEST)-GIVE-RESMIN(IBEST)) 67,67,53
   67 GIVE=DEN(IBEST)
   65 TORES=TORES-GIVE
      ICHECK=0
      GO TO 69
C
   53 IOLD=IBEST
      ICHECK=0
      JA=0
C     BLOCK 7 FLOW DIAGRAM
C     SELECT NEXT SMALLEST RATE QUALIFYING
```

APPENDIX A (CONTINUED)

```
      DO 71 K=LOW,NMAX
      IF(JSUCC(K)-1) 71,73,71
   73 ICHECK=ICHECK+1
      IF(K-IOLD) 72,71,72
   72 JA=JA+1
      IF(RATE(K)-SMALL) 79,74,74
   74 IF(JA-1) 71,79,71
   79 IF(TOTREQ(K)) 78,80,78
   80 IF(DAY-EF(K)) 84,82,82
   82 NODE=1
      JSUCC(K)=0
      JAMAX=JMAX-1
      IBEST=K
      JAP1=K+1
      GO TO 129
   84 JA=JA-1
      GO TO 71
   78 IF(ASSIGN(K)-RESMIN(K)) 83,81,81,
   81 IF(ASSIGN(K)-RESMAX(K)) 85,76,76
   85 GIVE=1.
      GO TO 87
   83 IF(RESMIN(K)-AVAIL(K) 89,89,76
   89 GIVE=RESMIN(K)
   87 IF(DEN(K)-GIVE-RESMIN(K)) 76,97,97
   76 JA=JA-1
      GO TO 71
   97 SMALL=RATE(K)
      IBEST=K
      IF(ICHECK-JMAX) 71,93,71
   71 CONTINUE
   93 IF(IBEST-IOLD) 47,94,47
   94 IF(DAY+1.-SMLF) 301,301,401
C     CALC NEW VALUES FOR ALL ACTS.
   69 DO 105 JA=LOW,NMAX
      IF(JSUCC(JA)-1) 105,107,105
  107 ICHECK=ICHECK+1
      IF(JA-IBEST) 111,109,111
  111 IF(TORES) 115,113,115
  115 IF(AVAIL(JA)-TORES) 117,117,119
  119 AVAIL(JA)=TORES
  117 TAVAIL(JA)=AVAIL(JA)+(ALF(JA)-DAY-1.)*GOOD(JA)
```

```
      GO TO 113
  109 ASSIGN(JA)=ASSIGN(JA)+GIVE
      TAVAIL(JA)=TAVAIL(JA)-GIVE
      DEN(JA)=DEN(JA)-GIVE
      AVAIL(JA)=AVAIL(JA)-GIVE
      ANTER(1)=DAY
      ANTER(2)=JA
      ANTER(3)=GIVE
      WRITE TAPE 10,ANTER
      IF(DAY-SMLF) 121,301,301
C     BLOCK 11
  121 IF(TAVAIL(JA)-DEN(JA)) 301,122,122
C     BLOCK 10
  122 IF(DEN(JA)) 113,123,113
C     BLOCK 10
  123 IF(DAY+1.-EF(JA)) 401,125,125
  125 JSUCC(JA)=0
      NODE=1
      JAMAX=JMAX-1
      JAP1=JA+1
  113 IF(ICHECK-JMAX) 105,127,105
  105 CONTINUE
  127 IF(NODE) 129,131,129
  129 ICHECK=0
C     BLOCK 12 FLOW DIAGRAM
C     INSURE THAT NO WORKING ACTS. HAVE J NODE = J OF COMPLETED ACT.
      DO 133 JA=LOW, NMAX
      IF(JSUCC(JA)-1) 133,135,133
  135 ICHECK=ICHECK+1
      IF(J(JA)-J(IBEST)) 137,139,137
C
  139 NODE=0
      JMAX=JAMAX
      GO TO 131
  137 IF(ICHECK-JAMAX) 133,141,133
  133 CONTINUE
C     FIND ACTS. THAT FOLLOW JUST COMPLETED ONE
  141 IF(JAP1-NMAX) 142,142,149
  142 DO 143 JA=JAP1,NMAX
      IF(I(JA)-J(IBEST)) 147,145,147
  147 IF(NODE) 143,149,143
  145 JAMAX=JAMAX+1
      JSUCC(JA)=1
      IF(TORES) 151,153,151
  151 IF(RESMAX(JA)-TORES) 155,155,157
```

APENDIX A (CONTINUED)

```
   155 AVAIL(JA)=RESMAX(JA)
       GO TO 159
   157 AVAIL(JA)=TORES
   159 IF(RESMAX(JA)-RAVAIL) 161,163,163
   161 GOOD(JA)=RESMAX(JA)
       GO TO 165
   163 GOOD(JA)=RAVAIL
   165 TAVAIL(JA)=AVAIL(JA)+(ALF(JA)-DAY-1.)*GOOD(JA)
   153 NODE=0
       DEN(JA)=TOTREQ(JA)
   143 CONTINUE
C
   149 JMAX=JAMAX
C      CHECK FOR FINISH PRIOR TO PRJDUR
   131 IF(JMAX) 132,134,132
   134 PRJDUR=DAY+1.
       GO TO 501
C      BLOCK 13
C      NOFINISH PRIOR TO ORIGINAL PRJDUR
   132 IF(TORES) 171,171,169
   171 GO TO (173,175,177),JFLAG
C      BLOCK 14 FLOW DIAGRAM
   173 DAY=DAY+1.
       ICHECK=0
   179 DO 181 JA=LOW,NMAX
       IF(JSUCC(JA)-1) 181,183,181
   183 ICHECK=ICHECK+1
       IF(RESMAX(JA)-RAVAIL) 187,185,185
   185 GOOD(JA)=RAVAIL
       AVAIL(JA)=RAVAIL
       GO TO 189
   187 GOOD(JA)=RESMAX(JA)
       AVAIL(JA)=RESMAX(JA)
   189 TAVAIL(JA)=AVAIL(JA)+(ALF(JA)-DAY-1.)*GOOD(JA)
       ASSIGN(JA)=0
       TORES=RAVAIL
       IF(ICHECK-JMAX) 181,169,181
   181 CONTINUE
C      CALC NEW RATES
   169 ICHECK=0
       DO 191 JA=LOW,NMAX
       IF(JSUCC(JA)-1) 191,193,191
   193 ICHECK=ICHECK+1
```

APPENDIX A (CONTINUED)

```
      IF(TOTREQ(JA)) 196,194,196
  194 RATE(JA)=0.
      GO TO 198
  196 RATE(JA)=TAVAIL(JA)/DEN(JA)
  198 IF(ICHECK-JMAX) 191,195,191
  191 CONTINUE
C     PICK NEW SMLF, SMALL RATE AND ALLOCATE
  195 NEWLOW=LOW
      GO TO 197
C     BLOCK 15 FLOW DIAGRAM
C     JFLAG=2, END .L.ALO .L. HI
  175 IF(DAY-ALO) 173,176,173
  176 REWIND 11
      ENZ=DAY
      IF(ENZ+1.-SMLF) 801,802,802
  801 ENZLO=DAY
  904 FORMAT(9H 175 DAY= ,F4.0)
      PRINT 904,DAY
      ICHECK=0
C     PUT NEW END VALUES ON TAPE 11
      DO 178 K=LOW,NMAX
      IF(JSUCC(K)-1) 178,180,178
  180 ICHECK=ICHECK+1
      FINIS(1)=K
      FINIS(2)=DEN(K)
      WRITE TAPE 11,FINIS
      IF(ICHECK-JMAX) 178,182,178
  178 CONTINUE
C     COMPLETED LISTING END INFO ON TAPE 11
  182 RAVAIL=RAVAIL-1.
      FINIS(1)=-1
      WRITE TAPE 11, FINIS
      JFLAG=1
      GO TO 173
C     BLOCK 16 FLOW DIAGRAM
C     JFLAG=3, END .L.HI .L. ALO
  177 IF(HI-DAY) 173,184,173
  184 REWIND 11
      ENZ=DAY
      IF(ENZ+1.-SMLF) 803,802,802
  803 ENZHI=DAY
```

APPENDIX A (CONTINUE)

```
      ICHECK=0
C     PUT NEW END VALUES ON TAPE 11
      DO 186 K=LOW,NMAX
      IF(JSUCC(K)-1) 186,188,186
  188 ICHECK=ICHECK+1
      FINIS(1)=K
      FINIS(2)=DEN(K)
      WRITE TAPE 11, FINIS
      IF(ICHECK-JMAX) 186,190,186
  186 CONTINUE
C     COMPLETED LISTING END INFO ON TAPE
  190 RAVAIL=RAVAIL+1.
      FINIS(1)=-1
      WRITE TAPE 11, FINIS
      JFLAG=1
      GO TO 173
  802 JA=KSML
      IBEST=KSML
      JSUCC(JA)=0
      NODE=1
      JAMAX=JMAX-1
      SMLF=SMLF+1.
      JAP1=JA+1
      PRINT 690,I(JA),J(JA),DAY,DAY
      GO TO 129
C     BLOCK 17 FLOW DIAGRAM
C     IF RAVAIL IS TOO LOW
  301 ICHECK=0
C     SET THE JSUCC OF PRESENT ACTS. C
      DO 303 JA=LOW, NMAX
      IF(JSUCC(JA)-1) 303,305,303
  305 JSUCC(JA)=0
      ICHECK=ICHECK+1
      IF(ICHECK-JMAX) 303,307,303
  303 CONTINUE
  307 ALO=DAY
      JMAX=0
      DAY=END+1.
      REWIND 11
      READ TAPE 11, FINIS
      K=FINIS(1)
      DEN(K)=FINIS(2)
```

APPENDIX A (CONTINUED)

```
      JSUCC(K)=1
      JMAX=JMAX+1
      LOW=K
      RAVAIL=RAVAIL+1.
  309 READ TAPE 11,FINIS
      IF(FINIS(1)+1.) 311,313,311
  311 K=FINIS(1)
      DEN(K)=FINIS(2)
      JSUCC(K)=1
      JMAX=JMAX+1
      GO TO 309
  313 ICHECK=0
C     BLOCK 18
      IF(ALO-HI) 315,317,317
C     BLOCK 19
  315 JFLAG=2
  317 GO TO 319
C     BLOCK 20 FLOW DIAGRAM
C     IF RAVAIL IS TOO HIGH
  401 ICHECK=0
C     SET THE JSUCC, OF PRESENT ACTS.=0
      IF(ENZLO-ENZHI) 402,404,402
  404 IF(ALO-DAY) 402,406,402
  406 JSUCC(IBEST)=0
      NODE=1
      JAMAX=JMAX-1
      JAP1=JA+1
      PRINT 609,I(IBEST),J(IBEST),ENZLO,DAY
      GO TO 127
  402 DO 403 JA=LOW,NMAX
      IF(JSUCC(JA)-1) 403,405,403
  405 JSUCC(JA)=0
      ICHECK=ICHECK+1
      IF(ICHECK-JMAX) 403,407,403
  403 CONTINUE
  407 HI=DAY
      JMAX=0
      DAY=ENZ+1.
      REWIND 11
      READ TAPE 11,FINIS
      K=FINIS(1)
      DEN(K)=FINIS(2)
```

```
      JSUCC(K)=1
      JMAX=JMAX+1
      LOW=K
      RAVAIL=RAVAIL-1.
  409 READ TAPE 11, FINIS
      IF(FINIS(1)+1.) 411,413,411
  411 K=FINIS(1)
      DEN(K)=FINIS(2)
      JSUCC(K)=1
      JMAX=JMAX+1
      GO TO 409
  413 ICHECK=0
C     BLOCK 21
      IF(HI-ALO) 415,319,319
C     BLOCK 22 FLOW DIAGRAM
  415 JFLAG=3
  319 REWIND 10
      SECEND=END+1.
  321 READ TAPE 10, ANTER
      IF(ANTER(1)-SECEND) 321,323,321
  323 BACKSPACE 10
      GO TO 179
C     BLOCK 24 FLOW DIAGRAM
C     WHEN ALL FINISHED, BEGIN PRINTING OUTPUT
  501 DAY2=1
      ANTER(1)=PRJDUR
      WRITE TAPE 10, ANTER
      END FILE 10
      REWIND 10
      PRINT 598
  502 JMAX=0
  504 READ TAPE 10, ANTER
      IF(DAY2-ANTER(1)) 508,506,508
C     FIND ACTS. RECEIVING RESOURCES, TOTAL FOR EACH AND SET JSUCC=1
  506 K=ANTER(2)
      IF(JSUCC(K)-1) 510,512,510
  510 JSUCC(K)=1
      JMAX=JMAX+1
      ASSIGN(K)=0.
  512 ASSIGN(K)=ASSIGN(K)+ANTER(3)
      GO TO 504
C     PRINT DAY, ACT AND ALLOCATED RESOURCES FOR 1ST ACT.
  508 BACKSPACE 10
```

APPENDIX A (CONTINUED)

```
      ICHECK=0
      DO 514 K=1,NMAX
      IF(JSUCC(K)-1) 514,516,514
  516 ICHECK=ICHECK+1
      KP1=K+1
      JSUCC(K)=0
      PRINT 600,DAY2,I(K),J(K),ASSIGN(K)
      TOTAL=ASSIGN(K)
      IF(ICHECK-JMAX) 517,524,524
  514 CONTINUE
C     PRINT REMAINING ACTS. AND ALLOCATE RESOURCES FOR DAY
  517 IF(KP1-NMAX) 518,518,524
  518 DO 520 K=KP1,NMAX
      IF(JSUCC(K)-1) 520,522,520
  522 ICHECK=ICHECK+1
      JSUCC(K)=0
      PRINT 602,I(K),J(K),ASSIGN(K)
      TOTAL=TOTAL+ASSIGN(K)
      IF(ICHECK-JMAX) 520,524,524
  520 CONTINUE
C     ALLFINISHED WITH PRINTING THIS DAY, PRINT TOTAL AND GO ON.
  524 PRINT 604,TOTAL
      DAY2=DAY2+1.
      IF(DAY2-PRJDUR) 502,580,580
C     FORMATS
    1 FORMAT(2F4.1)
    3 FORMAT(213,7F7.0)
  598 FORMAT(9HOFOR DAY ,F4.0,5X,10H ACTIVITY , 13,1H-,13,5X,10H REQUIRES
     1 ,F5.0,11H RESOURCES.)
  602 FORMAT(19X,9HACTIVITY ,13,1H-,13,5X,10H REQUIRES ,F5.0,11H RESOURC
     1ES.)
  604 FORMAT(67X,19HTOTAL FOR THIS DAY ,F6.0)
  690 FORMAT(10H ACTIVITY ,13,1H-,13,38H ENDED EARLY TO STOP LOOP BETWEE
     1N DAYS,F4.0,4H AND,F4.0)
C     WIND IT UP
  580 REWIND 10
      REWIND 11
      CALL EXIT
      END
```

APPENDIX B

COMPUTER PROGRAM VARIABLES

ALF(n)	Late finish of n^{th} activity
ALO	Day allocation procedure is stopped because of too few project resources/day
ALS(n)	Late start of n^{th} activity
ANTER(n)	Allocation information of n^{th} activity placed on tape.
ASSIGN(n)	Number of resources allocated to n^{th} activity on particular day of iteration
AVAIL(n)	Maximum number of resources/day that can be considered for allocation to n^{th} activity on day of iteration
DAY	Day of allocation
DAY2	Counter of keep track of days for printing
DEN(n)	Total resource required by n^{th} activity
EF(n)	Early finish of n^{th} activity
ENZ	Begining point for each iteration
ENZHI	Most recent date of a 'HI' (see 'HI')
ENZLO	Most recent date of a 'LO' (see 'LO')
ES(n)	Early start of n^{th} activity
FINIS(n)	Data about n^{th} activity placed on tape to allow program to back track to 'ENZ'
GIVE	Number of resources allocated to activity chosen for allocation

APPENDIX B (CONTINUED)

GOOD(N)	Permanent number of resources/day that could be allocated to nth activity
HI	Day procedure stopped because of too many project resources/-day available.
I	Subscript to identify activities
IBEST	Subscript of activity chosen for allocation
ICHECK	Counter to keep track of activities
IOLD	Last activity chosen as having smallest rate
J	Subscript to identify activities
JA	Same as ICHECK
JAMAX	Total number of activities qualifying for allocation consideration
JAP1	"JA" + 1
JFLAG	To signal Program of special requirement
JMAX	Same as 'JAMAX'
JSUCC(N)	Indicator that nth activity qualifies for allocation consideration
K	Subscript to identify activities
KP1	'K' + 1
KSML	Subscript of the activity with the smallest late finish at any particular moment.

APPENDIX B (CONTINUED)

LOW	Subscript for activity with lowest rate
LOWP1	'LOW'+ 1
N	Subscript to identify activities
NEWLOW	Subscript for last activity having smallest rate
NMAX	Counter to keep track of total number of activities
NODE	I node of activities being selected for consideration
PRJDUR	Project duration
RATE(n)	Necessity rate of n^{th} activity
RAVAIL	Number of project resources/day to be allocated on any one day.
RESMAX(n)	Maximum efficient size daily resource that can be allocated to n^{th} activity
RESMIN(n)	Minimum efficient size daily resource that can be allocated to n^{th} activity
SECEND	Variable used to relate 'DAY' to ENZ
SMALL	Smallest rate
SMLF	Smallest Late Finish
TAVAIL(n)	Total number of resources that activity (n) could receive from a particular moment until the late finish of that activity
TORES	Resources remaining to be allocated on day under consideration

APPENDIX B (CONTINUED)

TOTAL	Count of resources allocated on any particular day
TOTREQ(n)	Total number of resources required to complete n^{th} activity

BIBLIOGRAPHY

1 Brach, John R., "An Application of Resource Allocation Techniques," Department of Civil Engineering Thesis, M.I.T., 1963

2 Butler, David E., "A Computer Program for Balancing Man Power Requirements of Construction and Maintenance Projects," Department of Industrial Management Theses, M.I.T., 1961

3 Clough, Richard H., Construction Contracting, John Wiley & Sons, Inc., New York, 1960

4 Cutcliffe, J. Lloyd, "Lecture Notes on Critical Path Scheduling," October 1963

5 Farmer, Jack Roberts, Jr., "A Digital Computer Solution to the Critical Path Problem and an Approach to Resource Allocation," Department of Civil Engineering Thesis, M.I.T., 1962

6 "IBM General Information Manual FORTRAN" 1961

7 "IBM General Information Manual, PERT... a dynamic project planning and control method."

8 Keller, Joseph S., "The Critical Path Method - Its Implementation and Effective Utilization in Construction," Department of Civil Engineering Thesis, M.I.T., 1964

9 Kelley, James E., Jr., The Critical Path Method, Resources Planning and Scheduling, Mauchly Associates, Inc., 1961.

BIBLIOGRAPHY (CONTINUED)

10 Miller, C. L., Man-Machine Communications in Civil Engineering, M.I.T., 1963

11 Moore, Richard Stephen, "Resource Leveling in Operational Planning," Department of Civil Engineering Thesis, M.I.T., 1964

thesH757
A computer approach to resource allocati
3 2768 001 01521 7
DUDLEY KNOX LIBRARY